

UNITED STATES PATENT AND TRADEMARK OFFICE  
**CERTIFICATE OF CORRECTION**

PATENT NO. : 7,043,465 B2  
APPLICATION NO. : 09/876929  
DATED : May 9, 2006  
INVENTOR(S) : Patrick Pirim

Page 1 of 8

It is certified that error appears in the above-identified patent and that said Letters Patent is hereby corrected as shown below:

To the first page of the patent, in the Foreign Application Priority Data section, please add the following

-- Feb. 24, 2000 (FR) ..... 00 02355 --.

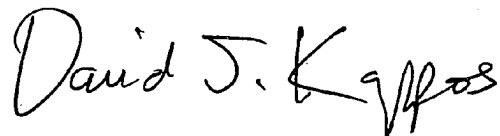
Column 2, lines 44-45, Column 27, line 15, and Column 29, line 33 replace "sizes of the associated frame" with

-- size of the associated frame --.

After the specification at Column 26, line 43, before the Claims, please add the attached Appendix A.

Signed and Sealed this

Twenty-third Day of November, 2010

A handwritten signature in black ink, reading "David J. Kappos". The signature is written in a cursive, flowing style with a large initial 'D' and 'K'.

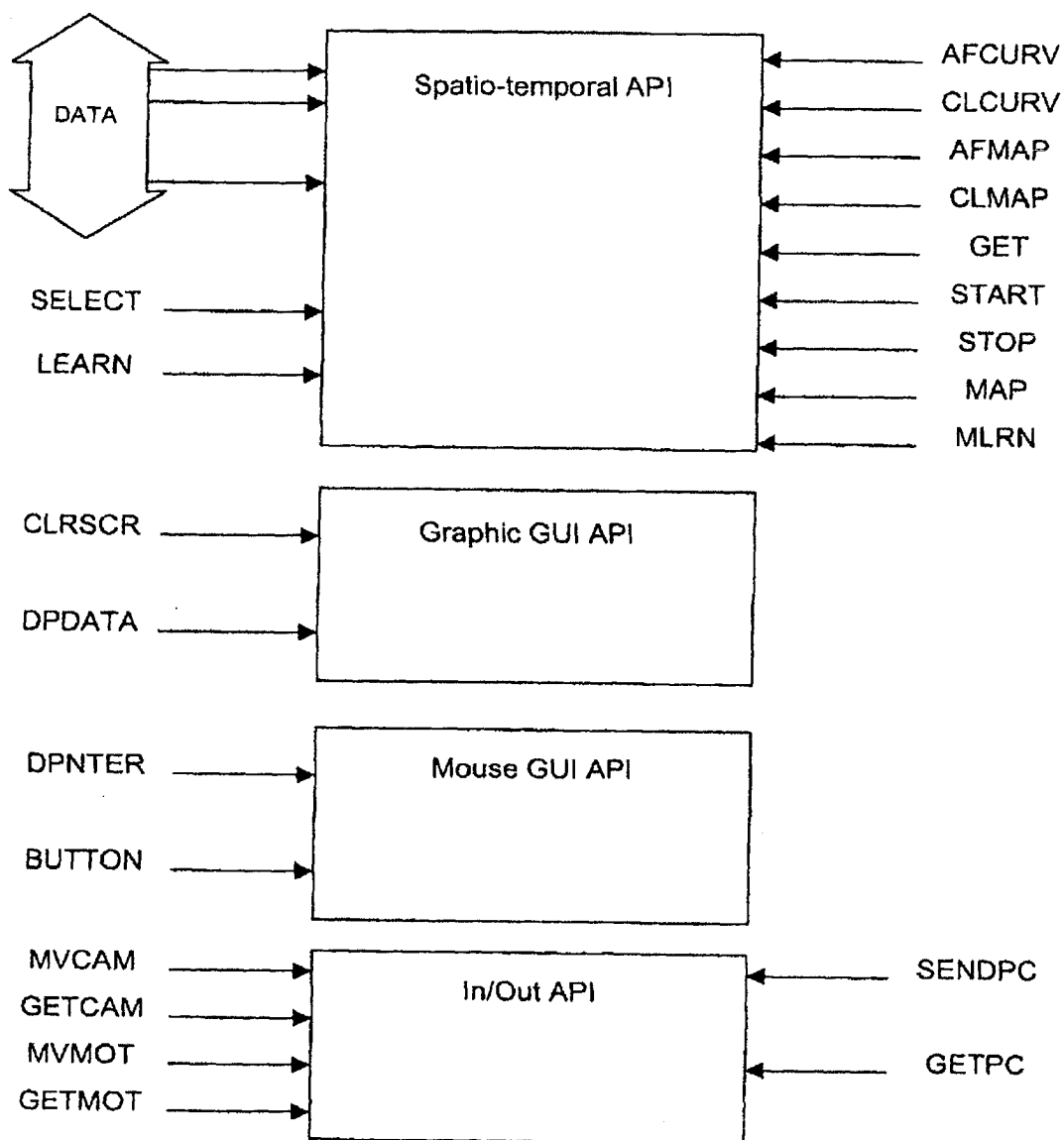
David J. Kappos  
*Director of the United States Patent and Trademark Office*

## APPENDIX A

### API Specifications

4 sub division for GVPP :

- Spatio-temporal computation API
- Graphic GUI API
- Mouse GUI API
- Communication and input-output API



Spatio-temporal API Bloc

This group enables all instructions to run the generic spatio-temporal computations and to get the results.

Functions :

**START :**

Goal: Initialisation of one bloc for the classification.

Parameter : index bloc, MIN value, MAX value.

Prototype :

Bloc3 equ 03  
MIN equ 10  
MAX equ 100

START Bloc3 MIN MAX

Input - R0 : index bloc  
R1 : MIN value  
R2 : MAX value

Output -

**STOP :**

Goal : end of computation.

Parameter : index bloc.

Prototype :

Bloc3 equ 03

STOP Bloc3

Input - R0 : index bloc  
Output -

**SELECT :**

Goal : Progammmation of input parameter bloc (lum, hue, motion, line orientation).

Parameter : Index bloc, type of input parameter.

Prototype :

Bloc3 equ 03  
LUM equ 00

SELECT Bloc3 LUM

Input - R0 : Index bloc  
R1 : Input parameter

Output-

**GET :**

Goal : Get the result computation of one parameter.

Parameter : Index bloc, Load result parameter.

Prototype :

```
Bloc3      equ  03
MIN        equ  00
MAX        equ  01
RMAX       equ  02
POSRMX     equ  03
POSMOY     equ  04
NBPTS      equ  05
.....
```

GET Bloc3 NBPTS

```
Input -    R0 :    Index bloc
           R1 :    Index parameter
Output-    R0 :    result value of this parameter
```

**LEARN :**

Goal : Learn the association-context of a bloc .

Parameter : Index bloc.

Prototype :

```
Bloc3      equ  03
```

LEARN Bloc3

```
Input -    R0 :    Index bloc
Output-
```

**MAP :**

Goal : Put on the time coincidences fonction the result of previous learning.

Parameter : Index bloc,summ of product-terms.

Prototype :

```
Bloc3      equ  03
```

MAP Bloc3 0F3 1AB 007

```
Input -    R0 :    Index bloc
           R1 :    First product terms
           R2 :    Second product terms
           R3 :    .....suite
Output-
```

**MLRN :**

Goal : Get the result of learning.

Parameter : Index Bloc.

Prototype :

MLRN			
Input -	R0	:	Index bloc
Output-	R0	:	MIN classification
	R1	:	MAX Classification
	R2	:	First main association (product terms)
	R3	:	Second association
	R4	:	.....suite

*AFCURV* :

Goal : Histogram curve drowing of one bloc.

Parameter : Index Bloc.

Prototype :

Bloc3	equ	03
AFCURV Bloc3		
Input -	R0	: Index bloc
Output-		

*CLCURV* :

Goal : Clear curve of one bloc.

Parameter : Index Bloc.

Prototype :

Bloc3	equ	03
CLCURV Bloc3		
Input -	R0	: Index bloc
Output-		

*AFMAP* :

Goal : Learning Bloc drowing.

Parameter : Index Bloc.

Prototype :

Bloc3	equ	03
AFMAP Bloc3		
Input -	R0	: Index bloc
Output-		

**CLMAP :**

Goal : Clear the learning bloc drawing.

Parameter : Index Bloc.

**Prototype :**

Bloc3 equ 03

CLMAP Bloc3

Input - R0 : Index bloc

Output-

**Graphic GUI API**

**CLRSCR :**

Goal : Clear Screen.

Parameter : No.

**Prototype :**

CLRSCR

Input -

Output-

**DPDATA :**

Goal : Display ASCII code on screen.

Parameter : ASCII code, row position, column position.

**Prototype :**

DPDATA

Input - R0 : ASCII code  
R1 : row position  
R2 : column position

Output-

**Mouse GUI API**

**DPNTER :**

Goal : Mouve and display the pointer.

Parameter : row position, column position.

**Prototype :**

DPNTER

Input - R0 : row position  
R1 : column position

Output-

*BUTTON :*

Goal : get the action of button.

Parameter : Button.

Prototype :

BUTTON

Input -

Output- R0 : new position of buttons

API E/S

*MVCAM :*

Goal : Move the camera.

Parameter : X Position, Y Position, Focus.

Prototype :

MVCAM

Input - R0 : X position

R1 : Y position

R2 : Focus

Output-

*GETCAM :*

Goal : Get the camera's parameters.

Parameter : No.

Prototype :

GETCAM

Input -

Output- R0 : X position

R1 : Y position

R2 : Focus

*MVMOT :*

Goal : Action motor.

Parameter : Sens+steps.

Prototype :

MVCAM

Input - R0 : Sens+steps

Output-

**GETMOT :**

Goal : Get the actual position of motor.  
Parameter : No.

Prototype :

GETMOT  
Input -  
Output- R0 : position

**SENDPC :**

Goal : Send one information to the PC.  
Parameter : information pointer.

Prototype :

SENDPC  
Input - R0 : information pointer  
Output-

**GETPC :**

Goal : Get an information from PC.  
Parameter : No.

Prototype :

GETPC  
Input -  
Output- R0 : information